Procedia Manufacturing

Volume 1, 2015, Pages 35–46

43rd Proceedings of the North American Manufacturing Research Institution of SME http://www.sme.org/namrc

Efficient Algorithms for Automated Inspection of Freeform Surfaces

Pradeep Biradar[1] and S. S. Pande[2]

[1,2]*Mechanical Engineering Department,*

Indian Institute of Technology Bombay, Mumbai - 400 076, India.

pvbiradar001@gmail.com, s.s.pande@iitb.ac.in

Abstract

This paper reports the development of efficient algorithms for automated inspection of freeform surfaces using point cloud data collected using an inspection system. The registration process to align the point cloud with the input CAD surface model is carried out in two phases, viz, rough registration followed by an accurate (fine) registration. For rough registration, algorithms based on the bounding box method and the convex hull method have been proposed. Fine registration is carried out using a computational geometry approach using a tessellated representation of the surface model. A random sampling technique has been integrated with the final registration algorithm to speed up the registration process. The algorithms have been compared with those reported in literature. Results show that the proposed algorithms substantially reduce the computational time, without sacrificing the accuracy of registration.

Keywords: Freeform surface, non-contact inspection, registration algorithms

1 Introduction

Freeform surfaces are widely used in several products such as automobile, aircraft, dies/molds and consumer goods. To meet stringent product quality requirements, it is important to assess if the manufactured surface lies within the specified tolerance limits. Inspection thus, forms a crucial part of the manufacturing process chain. Currently, non-contact laser inspection systems are becoming popular as they scan the entire part surface in a few seconds. The output of the inspection process is a 3D point cloud which is to be compared with the ideal part model. It is thus, very important to devise efficient algorithms for accurate and fast inspection of freeform surfaces. Literature reports several attempts to develop algorithms to accurately align point cloud data with the CAD model following two distinct approaches, viz, the point based method and the feature based method.

Peer-review under responsibility of the NAMRI Scientific Committee

doi:10.1016/j.promfg.2015.09.039

One important point based algorithm is the Iterative Closest Point (ICP) developed by (Besl and McKay, 1992) which computes the closest point on the CAD model for points on the measured surface using the quaternion approach. The algorithm terminates when the squared distance error between two consecutive iterations falls below a threshold value. (Arun et al., 1987) developed a registration algorithm based on the Singular Value Decomposition (SVD) method to register the data point set and the model point set. (Masuda and Yokoya, 1995) integrated a random sampling technique and a Median error estimator with the ICP algorithm to reduce its computational complexity. Their algorithm conducts several trials and chooses the one having the lowest Median error. (Chen and Medioni, 1992) used tangent plane approximations of the ideal surface to compute closest points, while (Ravishankar et al., 2010) developed a modified ICP algorithm using a high density of surface points refined over successive iterations. (Jing, 2014) utilized corner points of surface patches to construct coordinate systems and compute transformations required to match the coordinate systems.

Using the feature based registration approaches, (Krsek et. al., 1998) reported algorithm by extracting curves of zero mean curvature and matching based on similarity of lengths. (Chen and Bhanu, 2004) utilized the concept of Local Surface Patch (LSP) around each feature point. Properties of the LSPs are compared and the two surfaces are registered by matching centroids of corresponding LSPs. (He et al., 2013) extended this work using the Hausdorff curvature distances of LSPs for finding corresponding matches. (Gelfland et al., 2005) describe an algorithm which computes and uses a volume descriptor for surface points based on local geometry, for a coarse alignment. (Li and Gu, 2005) classified the model and the measured surfaces into distinct patches based on Gaussian and Mean curvature values and utilized patch properties to find corresponding matches. The centroids of the corresponding patches were used for registration. (Yamany and Farag, 1999) proposed a Surface signature matching (SSM) in which distance and angular features extracted from regions surrounding each point are represented using 2D signatures. The local match which produces the best registration between the surfaces is chosen for global registration.

Both point and feature based methods of registration have their own limitations. Point based methods normally require an initial approximate alignment between the point cloud and surface. Some of them employ computationally expensive minimization approaches for finding closest points on the surface. While feature based algorithms do not require any initial approximate correspondence between the two entities, the process of feature extraction requires a surface to be fitted to 3D point cloud. This process is again expensive. Surfaces which do not have large variations in surface properties cannot be efficiently handled by the feature based algorithms.

The algorithms proposed in this paper attempt to address these limitations without employing minimization approaches or extracting feature information from the point cloud. The main objective is to reduce computational complexity and speed up the registration process.

2 Overview of the system

The system developed in this work takes input as point cloud data and the 3D CAD model of the surface in the NURBS format. The process of registration consists of two stages, viz. an initial rough registration process to approximately align the point cloud with the surface followed by a final registration process for accurate alignment. Two algorithms have been proposed for rough registration, viz. the bounding box method and the convex hull method. Final registration has been carried out using tessellated representation of surface. In what follows, the rough and fine registration algorithms are discussed one by one.

2.1 Rough registration algorithms

The rough registration algorithms aim to establish a coarse alignment between the ideal NURBS based CAD model and the 3D point cloud (measured surface) without extracting surface features from the point data. The objective is to achieve rough registration at a fast speed without sacrificing the accuracy of registration as much possible. The two algorithms proposed are explained one by one.

Bounding Box approach

The Bounding Box method is based on the concept that since the point cloud and the parametric (CAD) surface represent the same object, the dimensions of the minimum volume axis parallel bounding boxes (MVAPBB) for them should be the same. MVAPBB has minimum volume, encloses the given entity completely and has faces parallel to XY, YZ, and ZX planes. Thus, by aligning the bounding boxes, the point cloud can be aligned with the parametric surface within some defined level of accuracy. The proposed algorithm utilizes the axes parallel, minimum volume bounding boxes of the point cloud and the surface for registering the two entities.

To compute the MVAPBB along the three principal axes, the point cloud is reoriented. An orthographic projection technique is used.
The steps involved in computing the bounding box are enumerated:

1. The centroid of the point cloud is computed and shifted to the origin.
2. The point cloud is projected onto one principal plane (say XY plane). The 2D axis parallel bounding box of the projected points and its area are computed. The point cloud is now rotated by a small angle about the axis perpendicular to the plane (Z axis say) and the bounding area is computed for the new orientation.
3. Orientation angle, at which the area of the bounding box is minimum, is computed using an optimization algorithm. Figure 1 shows the bounding box, its area and variation with the angle of rotation.

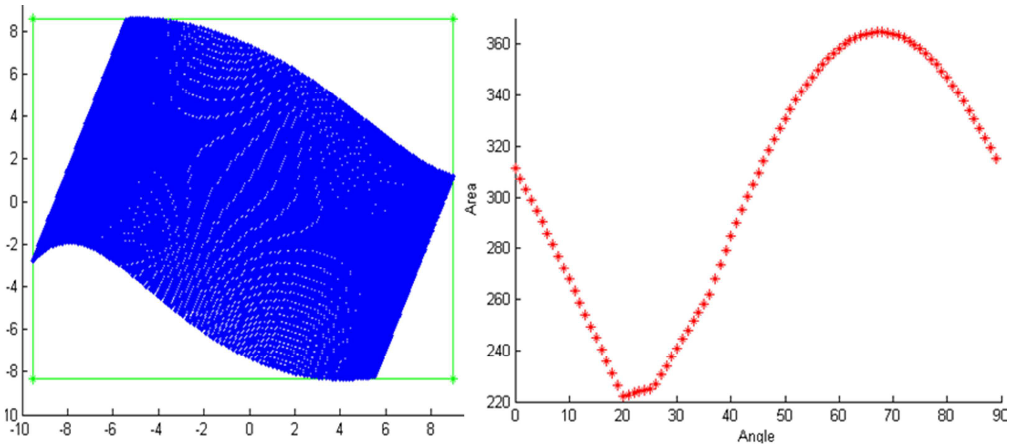


Figure 1: Bounding box area v/s Angle of Rotation θ

4. A 3D rotation matrix R, is constructed using the minimum area angle θ. 3D point cloud is reoriented using angle θ.

$$[P'] = [P] . [R] \tag{2.1}$$

Where,

[P] refers to the point coordinates P[x, y, z], R is rotation transformation matrix.

5. Steps 2, 3 and 4 are repeated to carry out the 3 rotations in order, viz., around Z, X and Y axes. A second set of rotations about the three principal axes is again carried out to take care of minor orientation corrections, if any.
6. The parametric equation of the CAD model NURBS surface is

$$P(u,v) = \sum_{i=1}^{m} \sum_{j=1}^{n} R_{i,j}(u,v) P_{i,j} \tag{2.2}$$

$$R_{i,j}(u,v) = \frac{B_{i,k}(u) B_{j,l}(v) w_{i,j}}{\sum_{i=1}^{m} \sum_{j=1}^{n} B_{i,k}(u) B_{j,l}(v) w_{i,j}}$$

Where,

$P(u,v)$ = Coordinates of point having parametric values u and v

m, n = number of control points along u, v parametric direction

$P_{i,j}$ = Control point

K,l = order of surface along u,v parametric direction

$B_{i,k}(u)$, $B_{j,l}(v)$ = B spline basis function along parametric directions

$W_{i,j}$ = weight assigned to control point

To compute the bounding box of the CAD model (NURBS surface), a few hundred sample points are selected on the surface using rectangular grid as well as random u,v values. Number of points generated should be at least 10-15% of the size of point cloud for good accuracy. The above procedure is applied to the generated surface points (CAD model) to compute the rotations required to reorient the surface. The rotations are applied to the control points sequentially to orient the surface such that its minimum volume bounding box will be parallel to the coordinate axes.

7. For rough registration the two bounding boxes (point cloud and model) are aligned by using the vertices of the bounding boxes.

For certain cases, the orientations of the point cloud and the surface may be flipped though their bounding boxes may properly align in (Figure 2). Such mismatches may arise if the point cloud has been rotated by 180 degrees with respect to the surface about one or more principal axis. A flip algorithm has been developed which detects such cases of mismatch in orientations by determining the principal axis about which the point cloud should be rotated by 180 degrees to properly align it with the surface. Details of algorithm are presented in (Biradar, 2014).

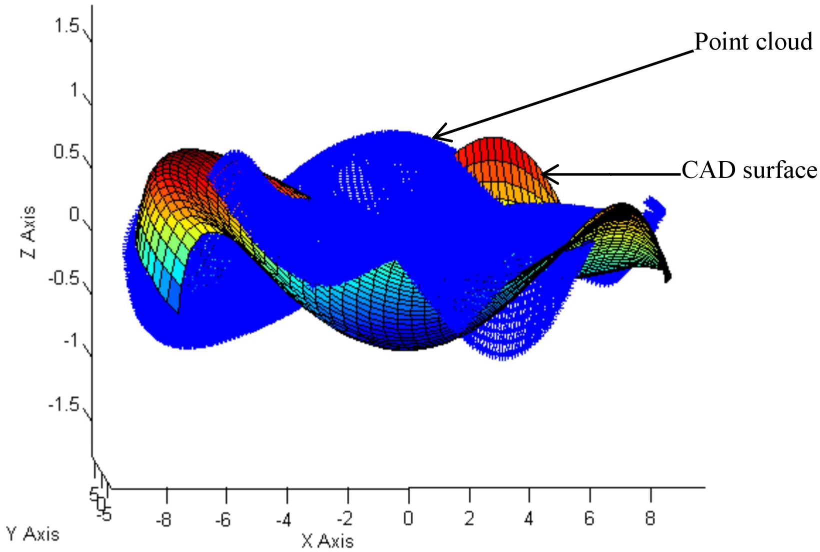


Figure 2: Flipped orientation of point cloud and the surface

Flip Algorithm

The flip algorithm uses orthographic projections to determine the axis about which the point cloud has to be rotated. Points of the point cloud and the points generated on the surface are projected onto a principal plane. Each projection is divided into 4 quadrants. By determining the correspondences between the quadrants, the axis of rotation for the point cloud is determined. Figure 3 shows the projection of point cloud and the surface on X-Y plane. Various cases of the mismatch are also shown.

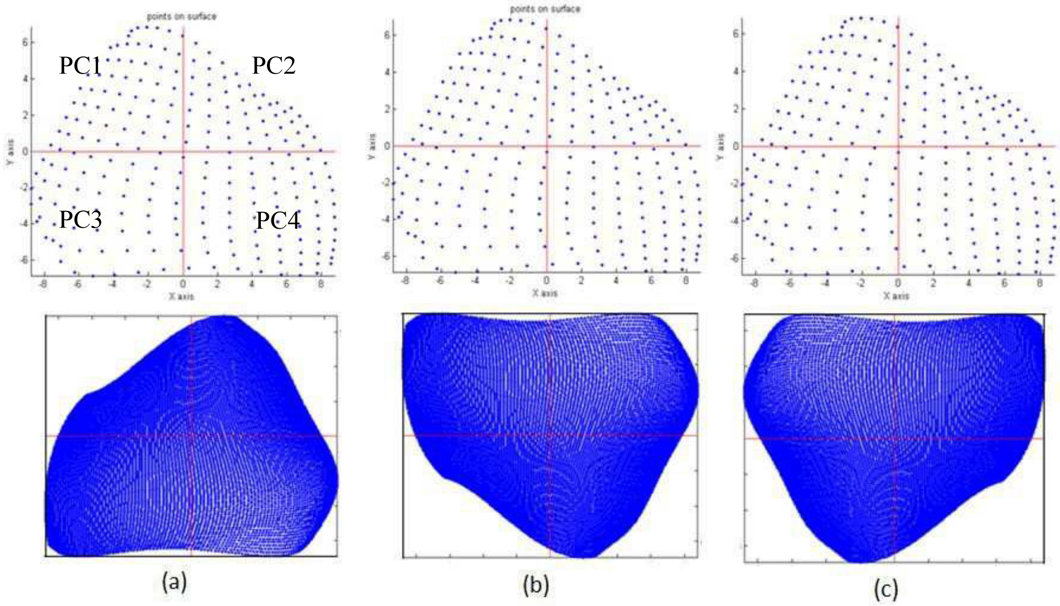


Figure 3: Selection of axis of rotation for flip algorithm

First quadrant of surface (P1) is compared with the 4 quadrants of point cloud (PC1 -4) to check the correspondence and decide the axis of rotation as under

Match	**Axis of Rotation**
P1→ PC1	No Rotation
P1→ PC2	Y (Figure 3a)
P1→ PC3	Z (Figure 3b)
P1→ PC4	X (Figure 3c)

To detect the correspondences between the quadrants of the point cloud and surface projections, area based approach have been proposed in this work. This approach finds correspondences between quadrants of point cloud and surface projections by comparing their areas. A 2D Delaunay Triangulation algorithm is implemented to triangulate the points in each quadrant, to get the area of the projection lying in the quadrant. The minimum area quadrants of the surface and point cloud projections are considered as corresponding matches. The rotations required to align the minimum area quadrants of the surface and point cloud are accordingly determined.

Convex Hull Approach

The convex hull method is based on identifying certain characteristic regions in the convex hulls of the point cloud and the CAD model (NURBS surface) and registering them by using local properties like surface normals of these regions. In this work, the quick hull algorithm (Carter et. al., 1996) has been used to compute the convex hull of the 3D point cloud.

Convex Hull of a surface is the smallest convex solid (minimum volume) which completely encloses the surface. A typical freeform surface may have convex, concave and flat regions. However, only flat and convex regions will have points which lie on the convex hull. Figure 4 shows the Free form surface and its convex hull. It is clearly seen that facets (triangles) having large area are found in concave regions and smaller area triangles are found in convex regions of the surface.

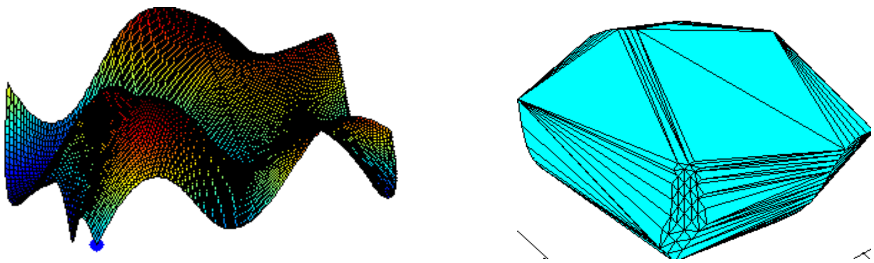
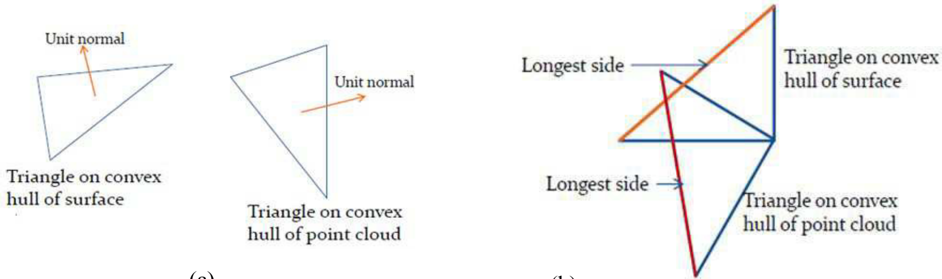


Figure 4: Freeform surface and Convex hull

Triangles in the concave regions are identified by way of triangles having area greater than a threshold value. Triangle having the largest area from the convex hull of the point cloud is determined and is paired with the corresponding largest area triangle of the convex hull of the surface. In case a number of triangles are having similar area, then angles between normals of each triangle and its neighbours are computed and are used as a reference value to find corresponding matches. From the corresponding pairs of matching triangles, a single pair is selected.

The normals of the matching pair of triangles and the angles between them are computed (Figure 5(a)). A rotation matrix is constructed using this computed angle. This matrix is used to rotate the point cloud to align the corresponding normals of the triangles.

After aligning the normals of the two triangles, the point cloud may still have a small angular displacement in the plane of the normal with respect to the surface. Since, the sides of the triangles will lie in the plane perpendicular to the normal, the longest sides of the corresponding pair of triangles are taken as a reference and the angle between them is determined as shown in Figure 5(b). The entire point cloud is rotated by this angle about the normal to align it with the surface. This completes the registration of the point cloud with the surface using the convex hull method. Convex hull method computes translations to align centroids of point cloud and the surface.



(a) (b)

Figure 5: Aligning normals of triangles (a) and longest sides of triangles (b)

2.2 Final registration algorithms

Rough registration methods do not establish specific correspondence between points/ features of the surface and the point cloud. Final registration algorithm aims to improve the accuracy of the rough registration established previously between the surface (CAD model) and the point cloud. Popular algorithms like the ICP attempt to minimize the residual distance error between the two entities using an iterative procedure which is computationally expensive. The minimization techniques require good starting points to converge rapidly and often have a tendency to get stuck in to a local minima. To address these issues it was proposed to carry out the final registration using an accurate tessellated representation of the CAD model (parametric surface). The algorithm is discussed below.

Registration using Tesselated surface representation

A) Tesselated surface generation

To construct the tesselated representation of the CAD model (parametric surface), an adaptive grid was generated on the parametric NURBS surface which was later triangulated.

The adaptive grid was generated as follows:

1. A uniform rectangular grid of points is generated on the parametric surface in the u, v parametric directions.
2. A threshold value of angular deviation of normal from Z-axis is fixed depending on the complexity of the surface. Typical values may be in the range if 1- 3 degrees.
3. The initial grid is refined alternately along the u or v parametric direction by considering two consecutive rows of points each time. If the angle between normals of corresponding points in the two rows is greater than the threshold value, a new row will be introduced. This procedure is continued till all corresponding points of adjacent rows satisfy the angular constraint.

Triangles are constructed by taking 3 adjacent points from the adaptive grid at a time. The coordinates of the vertices, the normal of the triangle and the equation of the plane of the triangle are stored. Th e relationship between grid points and triangles is stored separately which is utilized later for computation of closest points on triangles.

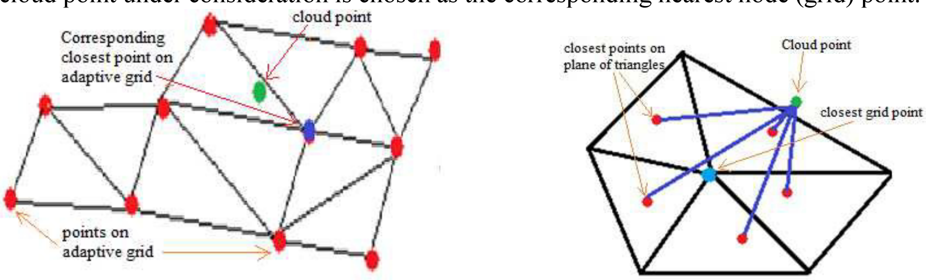
The generated tessellated surface is compared with the actual NURBS surface to check for deviations. Here, points are generated on the faces of triangles and their chordal distances from the surface are computed. If all distances lie within a predetermined threshold distance, the generated triangulated surface is deemed to be accurate enough and the procedure is terminated. If a few triangles have chordal distances greater than the threshold distance, then local re-triangulation is done by generating a new grid point on the surface.

B) Registration of point cloud with tessellated surface

After creation of a sufficiently accurate tessellated surface, the point cloud is registered with it using a 3 step iterative procedure. The steps are explained one by one.

1. Computing closest point on the adaptive grid

To compute the closest grid point on the tessellated model for a given cloud point, a small 3D window is constructed about the cloud point. The window will capture nearest grid points of the adaptive grid generated on the surface (Figure 6(a)). The grid point lying at the shortest distance from the cloud point under consideration is chosen as the corresponding nearest node (grid) point.



(a) (b)

Figure 6: Computation of closest point

2. Computing closest point on facet triangle

Triangles incident on the nearest grid point identified in the last step (1 ring topology), are identified as shown in Figure 6 (b). For a given cloud point, the closest point on the plane of each of the above candidate triangles is computed by dropping a normal from cloud point on the facet. Some of these points (foot prints) may lie inside the boundary of the corresponding triangles. Points lying outside the boundaries of the corresponding triangles are considered as false correspondences and are ignored. From this set, the closest point is computed and is selected as the final closest point on the triangulated surface.

3. Determining transformation parameters

After determining closest points on the tessellated surface for all the cloud points, the SVD algorithm (Arun et. al., 1987) is used to determine the transformation parameters required to register the cloud point with the surface. Point cloud P having N points and the corresponding set of closest points P′ on the tessellated surface are stored as 3xN matrices. Two matrices Q and Q' are constructed by shifting the centroids of P and P′ to the origin. H matrix is constructed as follows

$$H = Q \cdot (Q')^T \tag{2.3}$$

The SVD method is applied to the H matrix to compute its three constituent matrices U, Δ and V

$$H = U\Delta V^{T} \tag{2.4}$$

The rotation matrix to update the position of the point cloud is constructed as follows:

$$Rotation\ Matrix\ (X) = VU^{T} \tag{2.5}$$

If determinant of the rotation matrix is equal to 1, then the above matrix is applied to the point cloud to update its position. At the end of the iteration, the squared distance error term, i.e., the sum of squared Euclidean distances between points of the point cloud and corresponding closest points on the tessellated surface is determined. This completes one iteration of the algorithm.

Several such iterations are carried out. From the second iteration onwards, the squared distance error term of the current iteration is compared with the term from the previous iteration. If the difference between the terms lies below a threshold (typical value is 1x10−5), the iterative process is terminated. This marks the completion of the final registration process of the point cloud with the CAD model.

Registration with random sampling

To reduce the computational time of the tessellated surface algorithm for registration, a random sampling technique has been proposed. A Least Median (LM) error estimator term is defined for this purpose. Initially, a random sample of points is chosen from the 3D point cloud. The squared distance error between each point of the random sample and the corresponding closest point on the surface model $P(x', y' z')$ is computed as follows:

$$\text{Squared distance error term} = [(x' - x)^2 + (y' - y)^2 + (z' - z)^2] \tag{2.6}$$

The median value of these squared distance error terms is chosen as the Median Error Estimator. The steps of the random sampling algorithm are as under:

1. For each trial, initially a random sample of points is selected from the point cloud.
2. For each sample point, the closest point on the tessellated surface is determined by first computing the closest point on the tessellated surface adaptive grid. The true closest point on the face of these triangles is computed next.
3. Transformation parameters required to align the sample points with their corresponding closest points on the triangulated surface are computed using the SVD algorithm.
4. Transformation parameters are applied to the sample points to update their positions.
5. The squared distance error term is computed for the sample points after updating their positions.
6. Steps 2, 3, 4 and 5 are repeated iteratively for each trial until the difference in the squared distance error terms of successive iterations falls below a given threshold.
7. The median estimator error term is computed at the end of each trial.
8. Steps 1to 7 are repeated for new trials. If the median estimator error term of the new trial is smaller than the median estimator term of the preceding trial, then transformations computed during the new trial are sequentially applied to the point cloud to update its position.
9. At the end of the pre-decided number of trials, the point cloud is considered to be accurately registered with the surface.

3 Results and discussion

All the rough and fine registration algorithm enumerated above were implemented in MATLAB version R2011A on the PC system (Intel® Core™ i5-2300 processor, 3.23 GB RAM). The performance of these algorithms was compared in terms of speed and accuracy with the algorithms reported in literature. Li's surface curvatures algorithm (Li and Gu, 2005) was used to benchmark the performance of the proposed Bounding Box and Convex Hull rough registration algorithms, while the ICP algorithm integrated with Least Median (LM) estimator (Masuda and Yokoya, 1995) was used for comparison for the tessellated fine registration algorithm. Algorithms were rigorously tested for several case studies of CAD model surfaces with varying geometric complexities, random point locations and errors of varying magnitudes to generate the simulated point cloud (Biradar, 2014). Results of a typical case study are presented here.

Case study

Inputs provided to the system include a bi-cubic NURBS surface (CAD model) designed with 36 control points, given in Table 1.

(0,0,0)	(4,0,2)	(8,0,-1)	(12,-4,1)	(16,-1,2)	(20,0,1)
(-1,2,1)	(4,3,2)	(8,4,4)	(12,5,-1)	(16,4,2)	(22,3,-1)
(1,8,-2)	(4,8,4)	(8,9,-3)	(12,7,0)	(16,8,1)	(20,9,-1)
(1,12,1.5)	(4,12,3)	(8,12,0)	(12,11,1)	(16,10,2)	(20,13,3)
(-2,16,-2)	(4,15,-1)	(8,14,-2)	(12,15,2)	(16,16,1)	(18.5,14,0)
(0,20,1)	(4,21,-1)	(8,23,2)	(12,16,2)	(16,20,1)	(18,17,0)

Table 1: Control points of NURBS surface

The knot vector used for the surface is Open Uniform with knot values [0 0 0 0 0.33 0.66 1 1 1 1]. Figure 4 shows the NURBS design surface CAD model. An ideal point set having 10000 points was generated from the surface. Random deviations were added to the point set to simulate the point cloud obtained by the non-contact inspection. This point cloud was used for testing registration with the surface model.

The performance of the rough registration algorithms was tested under different initial random orientations of the point cloud. A deviation of ± 0.010 mm was added to the ideal points of the point set along X, Y and Z directions. 1200 points were generated on the surface and were used to compute the bounding box and convex hull of the surface. The inputs used for different trials are given in Table 2. Table 3 shows the results of three rough registration algorithms for different trials of case study.

Trial No:	Initial rotations(degrees)			Initial translations(mm)		
	About z axis	About y axis	About x axis	X	Y	Z
1	28.21	16.52	26.84	0.25	0.18	0.12
2	44.68	26.89	13.65	0	0	0
3	53.41	25.72	37.89	0.26	0.51	0.83

Table 2: Inputs data for rough registration

Trial No	Deviations of cloud points from surface (mm)					
	Bounding Box		Convex Hull		Li's Surface Curvature (Li and Gu, 2005)	
	Max	Avg.	Max	Avg.	Max	Avg.
1	0.0630	0.0234	0.0641	0.0234	0.0213	0.0112
2	0.0802	0.0297	0.0216	0.0107	0.0212	0.0108
3	0.0716	0.0274	0.0205	0.0104	0.0199	0.0101

Table 3: Comparison of rough registration algorithms

The runtimes for the rough registration algorithms for Trial 1 are Surface curvatures algorithm (1690 seconds), Bounding box algorithm (172.3 seconds) and Convex hull algorithm (171.2 seconds)

Results, (Table 3) show that the bounding box method has the least accuracy of registration, while highest accuracy is provided by Li's surface curvature algorithm. The accuracy of the convex hull method is very close to Li's surface curvatures method. Both the bounding box and the convex hull algorithms are approximately 10 times faster than the Li's surface curvatures algorithm. The convex hull algorithm thus, looks very promising in terms of both speed and accuracy of rough registration.

To compare the performance of the tessellated final registration algorithms, different simulations were carried out with random deviations of 10, 20 and 30 microns added to the point cloud. The convex hull method was used for rough registration. 400 points were generated on the ideal surface model for computing the convex hull. Performance of algorithms involving random sampling, viz, ICP integrated with LM estimator and tessellated surface integrated with random sampling were compared. (Table 4)

Trial No:	No. of Sample Points	No. of Trials	Deviation in point cloud (microns)	Deviation of cloud points from surface after registration (microns)					
				Tessellated surface		Tessellated surface with random sampling		ICP algorithm with LM estimator	
				Max	Avg.	Max	Avg.	Max	Avg.
1	30	10	10	18.50	0.3149	32.8	0.1111	30.2	0.1898
2	40	8	20	20.1	0.4189	38.4	0.3694	35.20	0.2654
3	30	12	30	20.7	0.6610	34.6	0.4648	31.3	0.5529

Table 4: Performance of fine registration algorithms

The runtimes for the final registration algorithms for Trial No:1 are Tessellated surface (338.2 seconds), tessellated surface with random sampling (14.5 seconds) and ICP algorithm with LM estimator (2051.6 seconds)

It is clearly seen that the runtime of the tessellated surface algorithm integrated with random sampling is the least. This algorithm is approximately 140 times faster than the ICP algorithm integrated with LM estimator. The maximum deviations reported by the ICP algorithm and the tessellated surface random sampling algorithm are almost similar, which indicates that the algorithms have similar accuracy. Tessellated surface random sampling algorithm thus, looks very promising in terms of both speed and accuracy for fine registration.

4 Conclusions

The rough registration algorithms proposed in this work do not utilize any surface information like curvatures to register the point cloud with the surface, and can thus, handle both simple and complicated surfaces. The convex hull rough registration algorithm is marginally less accurate than Li's surface curvatures algorithm but is about 10 times faster. For final registration, Tessellated surface representation with random sampling substantially decreases the computation time without sacrificing the accuracy of registration. In conclusion, convex hull algorithm for rough registration and Tessellated surface random sampling algorithm for fine registration are very promising in terms of speed and accuracy for the inspection of freeform surfaces.

References

Arun K.S., Huang T.S., Blostein S.D. (1987). Least-Squares Fitting of Two 3-D Point Sets. *IEEE transactions on pattern analysis and machine intelligence*, 14(2), 698-700.

Besl P., McKay N. (1992). A method for registration of 3D shapes. *IEEE transactions on pattern analysis and machine intelligence* , 14(2), 239-256.

Biradar P. (2014). Computer Aided Inspection of Freeform Surfaces *M. Tech. Thesis.* Indian Institute of Technology Bombay, India.

Carter C. B., Dobkin D. P., (1996). Huhdanpaa H. The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software* , 22 (4), 469-483.

Chen H., Bhanu B. (2004). 3D Free-Form object recognition in range images using local surface patches. In: *Proc. of the 17th ICPR'04, 1051- 4651/04.*

Chen Y., Medioni J. (1992). Object modelling by registration of multiple range images. *Image Vision Comput* ,10(3), 145-155.

Gelfand N., Mitra N. J., Guibas L. J., Pottmann H. (2005). Robust Global Registration. *Eurograhics Symposium on Geometry Processing.*

He B., Lin Z., Li Y. F. (2013). An automatic recognition algoritm for the scattered point clouds based on the curvature feature. *Optics and Laser Technology*, 46, 53-60.

Jing. (2014). Calculation of profile error for complex surface. *Measurement*, 48, 183-186.

Krsek P., Pajdla T., Hlavac V., Martin R. (1998). Range Image registration driven by a hierarchy of surface differential features. *22nd workshop of the Austrian association for pattern recognition,* 175-183.

Li Y., Gu P. (2005). Inspection of free-form shaped parts. *Robotics and Computer-Integrated Manufacturing*,21, 421-430.

Masuda T., Yokoya N. (1995). A robust method for registration and segmentation of multiple range images. *Computer vision and image understanding*, 61(3), 295-307.

Ravishankar S., Dutt H. N. V., Gurumoorthy B. (2010). Automated inspection of aircraft parts using a modified ICP algorithm. *Int. J. Adv. Manuf. Technol ,* 46, 227-236.

Yamany S. M., Farag A. A. (1999). Free-form surface recognition using surface signatures. *Int. Conf. on Computer Vision ,* 2, 1098-1104.